

# CRYPT-02

To hide passwords, always use with a random salt for each hash

Sean Barnum, Cigital, Inc. [vita<sup>1</sup>]

Copyright © 2007 Cigital, Inc.

2007-03-22

## Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 3822 bytes

Attack Category	<ul style="list-style-type: none"><li>Encryption Assault</li></ul>								
Vulnerability Category	<ul style="list-style-type: none"><li>Cryptography</li></ul>								
Software Context	<ul style="list-style-type: none"><li>Cryptography</li></ul>								
Location									
Description	<p>If crypt() must be used to hide passwords, it should be used with a random salt for each hash.</p> <p>The crypt() function should be avoided in favor of stronger hash functions. However, if crypt() must be used, it should never be used in the form crypt(pwd, pwd), where the password is also being used as the salt. This is because the first two bytes of the salt will always be prepended to the hash and will be visible in plaintext. Hence, using the password as the salt reveals two characters of the password. This leaves only 6 characters of "secret information," which would require a dictionary of size less than 2^40 to crack. Also, if the same salt is used for all passwords, then once an attacker knows the salt, his or her work is reduced to the same amount as when no salt was present.</p>								
APIs	<table><tr><th>Function Name</th><th>Comments</th></tr><tr><td>crypt</td><td>look for crypt(x, x) with same param twice</td></tr></table>			Function Name	Comments	crypt	look for crypt(x, x) with same param twice		
Function Name	Comments								
crypt	look for crypt(x, x) with same param twice								
Method of Attack									
Exception Criteria									
Solutions	<table><tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr><tr><td>When one must use crypt()</td><td>The ideal solution is to use a more secure routine provided by a cryptographic library. If this</td><td>Somewhat effective. Use of a more secure hash algorithm would be preferable.</td></tr></table>			Solution Applicability	Solution Description	Solution Efficacy	When one must use crypt()	The ideal solution is to use a more secure routine provided by a cryptographic library. If this	Somewhat effective. Use of a more secure hash algorithm would be preferable.
Solution Applicability	Solution Description	Solution Efficacy							
When one must use crypt()	The ideal solution is to use a more secure routine provided by a cryptographic library. If this	Somewhat effective. Use of a more secure hash algorithm would be preferable.							

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

	is not possible, ensure that the salt changes with each password and that it cannot be computed using the password. The salt and the password must be completely independent of each other.	
<b>Signature Details</b>	<code>char *crypt(const char *pwd, const char *pwd)</code>	
<b>Examples of Incorrect Code</b>	<code>hash = crypt(password, password);</code>	
<b>Examples of Corrected Code</b>	<pre>salt = get_random_salt(); /* See McGraw et al. p. 345 for a sample implementation of this function. */ hash = crypt(password, salt);</pre>	
<b>Source Reference</b>	Viega, John & McGraw, Gary. Building Secure Software: <i>How to Avoid Security Problems the Right Way</i> . Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, pp. 337+. This is a good description of crypt().	
<b>Recommended Resource</b>		
<b>Discriminant Set</b>	<b>Operating Systems</b>	<ul style="list-style-type: none"> <li>• UNIX (All)</li> <li>• Windows</li> </ul>
	<b>Languages</b>	<ul style="list-style-type: none"> <li>• C</li> <li>• C++</li> </ul>

## Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at [copyright@cigital.com](mailto:copyright@cigital.com)<sup>1</sup>.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@cigital.com>